



PATENT  
Attorney Docket No.: 40101/07301

AF 12178  
ZPW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: )  
Mark A. Stevens ) Group Art Unit: 2178  
Serial No.: 09/854,111 ) Examiner: Cong-Lac T Huynh  
09/766,335 )  
Filed: January 19, 2001 ) Board of Patent Appeals and  
For: CONVERSION SYSTEM FOR ) Interferences  
TRANSLATING STRUCTURED )  
DOCUMENTS INTO MULTIPLE )  
TARGET FORMATS )

Mail Stop: Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37**

In support of the Notice of Appeal filed April 4, 2006, and pursuant to 37 C.F.R. § 41.37, Appellant presents their appeal brief in the above-captioned application.

This is an appeal to the Board of Patent Appeals and Interferences from the Examiner's final rejection of claims 1, 4-20 and 22-38 in the final Office Action dated December 5, 2005. The appealed claims are set forth in the attached Claims Appendix.

06/05/2006 HNGUYEN1 00000103 09854111

01 FC:1402

500.00 OP

1. Real Party in Interest

This application is assigned to Wind River Systems, Inc., the real party in interest.

2. Related Appeals and Interferences

There are no other appeals or interferences which would directly affect, be directly affected, or have a bearing on the instant appeal.

3. Status of the Claims

Claims 1, 4-20 and 22-38 were rejected in the Final Office Action dated December 5, 2005. The final rejection of claims 1, 4-20 and 22-38 is being appealed.

4. Status of Amendments

All amendments submitted by the appellants have been entered. None were submitted after the Advisory Action.

5. Summary of Claimed Subject Matter

The present invention as recited in claim 1 is directed to a translator for translating a source file in a source format to a target file in a target format. (*See Specification*, p. 7, line 36 – p. 8, line 7). The translator comprises a feature identifier to determine a feature set of the source file. (*See Id.*, p. 8, lines 15-18). The translator further comprises a buffer to assemble the feature set. (*See Id.*, p. 8, line 33 – p. 9, line 5). The translator further comprises a

feature writer to write the feature set into the target file in the target format. (*See Id.*, p. 9, lines 5-10).

The present invention as recited in claim 20 is directed to a method of translating a file from a source format to a target format, the method comprising identifying a feature set of a source file, assembling the feature set in a buffer, and writing the feature set into a target file in the target format. (*See Id.*, p. 5, line 37 – p. 6, line 8).

The present invention as recited in claim 34 is directed to a method of configuring a system to translate a source file in a source format to a target file in a target format, the method comprising providing a feature identifier to determine a feature set of the source file, providing a buffer to assemble the feature set, and providing a feature writer to write the feature set into the target file in the target format. (*See Id.*, p. 6, lines 9-14).

The present invention as recited in claim 35 is directed to a system for translating a source file in a source format to a target file in a target format, the system comprising a feature identifier to determine a feature set of the source file, a buffer to assemble the feature set, and a feature writer to write the feature set into the target file in the target format. (*See Id.*, p. 6, lines 15-19).

The present invention as recited in claim 36 is directed to an article of manufacture for translating a source file in a source format to a target file in a target format, the article of manufacture comprising a computer usable medium having a computer readable program code embodied therein, the computer usable medium having computer readable program code for identifying a feature set of the source file, computer readable program code for assembling the feature set, and computer readable program code for writing the feature set into the target file in the target format. (*See Id.*, p. 6, lines 19-27).

The present invention as recited in claim 37 is directed to computer readable

program code for translating a source file in a source format to a target file in a target format, the computer readable program code comprising computer readable program code for identifying a feature set of the source file, computer readable program code for assembling the feature set, and computer readable program code for writing the feature set into the target file in the target format. (*See Id.*, p. 6, lines 28-34).

The present invention as recited in claim 38 is directed to a translator for translating a source file in an MIF format to a target file in an HTML format, the translator comprising a feature identifier having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file, a buffer to store the feature set, and a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format. (*See Id.*, p. 13, line 33 – p. 14, line 9).

6. Grounds of Rejection to be Reviewed on Appeal

I. Whether claims 1, 4-20 and 22-38 are unpatentable under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Pub. No. 2002/0052893 to Grobler et al. (“the Grobler publication”).

II. Whether claim 38 is unpatentable under 35 U.S.C. § 103(a) as obvious over the Grobler publication.

7. Argument

I. The Rejection of Claims 1, 4-20 and 22-38 Under 35 U.S.C.  
§ 102(e) as Being Anticipated By U.S. Patent Application  
Pub. No. 2002/0052893 to Grobler et al. Should Be Reversed.

A. The Examiner's Rejection

In the final Office Action, the Examiner rejected claims 1, 4-20 and 22-38 under 35 U.S.C. § 102(e) as being anticipated by the Grobler publication. (*See 12/05/05 Office Action*, p. 2, lines 22-23).

The Grobler publication generally describes a computer-based method and system for importing a table data from a selected source document into a selected target document. (*See, Grobler publication, Abstract*). Specifically, a user of the system may select a source and a target during the import process. (*See Id., ¶¶ 0060-0061*). The selections may be using either a drag-and-drop function or by a copy-and-paste function of a clipboard provided by the system. (*See Id., ¶¶ 0019, 0062*). Upon selecting the source and target of the import process, the selected source data is temporarily stored within the system. (*See Id., ¶ 0063*). The table structure of the temporarily stored source data by parsing the source data for tags in order to identify columns and rows contained in the source data, in addition to the contents of the columns and rows of the source data. (*See Id., ¶ 0064*). After the table structure of the source data is analyzed, the system determines whether the user has selected to specify the format of the target table, wherein the user may choose and modify the data contained in the source data. (*See Id., ¶¶ 0068-0069*).

After processing the user settings, the system creates an empty target table having a table structure in accordance with the results of the user settings. (*See Id., ¶ 0070*). Finally, the empty table target is filled with the source data by inserting the contents of the columns and rows of the source data into the empty target table. (*See Id., ¶ 0072*).

B. The Grobler Publication Does Not Disclose “(a) Identifying a Feature Set of a Source File; (b) Assembling the Feature Set in a Buffer; and (c) Writing the Feature Set into a Target File in the Target Format” as Recited In Claim 20.

The Examiner has stated the grounds for rejection with respect to claim 20, and as such, Applicant will direct the initial response to that claim. The Examiner asserts that the Grobler publication discloses identifying a feature set of the source file, assembling the feature set in a buffer, and writing the feature set into a target file. (*See 12/05/05 Office Action*, p. 2, line 25 – p. 3, line 9). However, this assertion is incorrect. To anticipate the identifying step of the method recited in claim 20 of the present application, the Examiner relies on the “tags” of the source data described in the Grobler publication. (*See Id.*) According to the Grobler publication, the system parses the source data for *tags* that indicate the table structure of the source data in order to identify the columns, rows, and contents contained in the source data. (*See, Grobler publication, ¶ 0064*). The tags of the Grobler publication are presented in tag language and are described as “plain text commands preceding the content to be displayed to indicate the format in which the content should be displayed.” (*See Id., ¶ 0005*). A tag language application is any software program that is capable of accessing and handling files written in a tag language, such as HTML and rich text format (“RTF”). (*See Id., ¶ 0040*). Both HTML and RTF provide information on the text through the use of tags, or commands, being located in front of the data. (*See Id.*). Thus, the parsing of the source data for tags as described by the Grobler publication is simply the process of indicating a plurality of commands located in front of various pieces of contents within the source data.

In contrast to the Grobler publication, claim 20 of the present invention recites, “identifying a feature set of a source file.” The identification of a set of features is not equivalent

to parsing source data for tags. The definition for features includes paragraph styles, straddled cells in a table, cross referencing, pen styles in a drawing, order indicators, location indicators, etc. and combinations thereof. (*See Specification*, p. 8, lines 18-28). Features may be used in conjunction with a graphic to indicate location, format, and the data of the graphic to define a graphic feature group. (*See Id.*, p. 20, lines 23-34). A translator according to the present invention may read through a source file to collect information about each of these sets of features and feature groups. (*See Id.*, p. 8, lines 18-28). As opposed to indicating the tags, or commands, within the source data as described in the Grobler publication, the present invention identifies a set of features that may be represented by not just a command, but also by a parameter of a command, or by multiple commands spread throughout the source file. (*See Id.*, p. 8, lines 28-32). Rather than parsing for individual commands, the present invention identifies feature parts, or feature sets, within a source file. By being feature directed, the present invention is more generically applicable and is less closely tied to any particular file format than the command directed system of the Grobler publication. Furthermore, this identifier based on feature sets advantageously allows the present invention to use a back-end lookup table that is flexible enough to write nominally distinct tree structured formats, sequential formats, post-fix formats and other discrete file format structures. (*See Id.*, p. 11, lines 6-10). Thus, the identifying a feature set of a source file is not equivalent to parsing source data in order to indicate tags for identifying columns, rows, and contents.

In addition, claim 20 of the present application recites, “assembling the feature set in a buffer.” To anticipate the assembling step of the method recited in claim 20 of the present application, the Examiner relies on the selection and temporary storage of the tags in a “clipboard” described in the Grobler publication. (*See 12/05/05 Office Action*, p. 2, line 25 – p.

3, line 9). However, this is not equivalent to *assembling* the feature set in a buffer. According to the present invention, after identifying a description of the feature set, the present invention assembles the feature set in an intermediary storage area such as a buffer. (*See Id.*, p. 8, lines 33-36). As discussed above, the feature set may include information related to the order, or sequence, of the features. This buffer serves as a convenient component for retaining the feature set information for assembling the set, specifically in the event the sequence of the features within the set needs to be reorganized. (*See Id.*, p. 11, lines 31-35). An example of a set of features that may require reorganization would be document header and footer specifications. (*See Id.*, p. 17, lines 7-24). The header, body, and footer of a document are typically written in sequential order with the header being first. However, if some of the information written to the header is not found until the entire document has been read, then the header may be identified and stored last in the buffer. In order to write the header, body, and footer in the proper order, the present invention assembles this feature set in the buffer. (*See Id.*). Thus, in a format where the feature order is important in a target format, the present invention allows for the features of a set to be arranged in the buffer in a specific order relative to each of features. The Grobler applicant fails to teach or suggest, “assembling the feature set in a buffer,” as recited in claim 20. As discussed above, the Grobler publication merely discusses filling the target table with columns, rows, and contents of the source table. The clipboard simply provides a temporary storage for a “copy and paste” function. (*See, Grobler publication, ¶ 0036*). After the “copy” function and before the “paste” function, the only disclosed operation performed on the temporarily stored data is the parsing analysis for the indication of tags. (*See Id.*). Indicating tags in the source data is entirely different from *assembling* the feature set. The Grobler publication does not disclose any further operations performed on the source data while the data

is stored on the clipboard. It is neither implied nor inherent in the Grobler publication that the process of copying source data to a temporary storage for analysis would further include the step of *assembling* or reorganizing or rearranging. While the Applicant maintains that the feature set of the present invention is distinct from the tags of the Grobler publication, it is further submitted that the Grobler publication does not perform an equivalent function to the “assembling the feature set in a buffer,” as recited in claim 20.

In view of the above arguments, it is respectfully submitted that the Grobler publication fails to disclose or suggest a “method of translating a file from a source format to a target format, the method comprising: (a) identifying a feature set of a source file; (b) assembling the feature set in a buffer; and (c) writing the feature set into a target file in the target format,” as recited in claim 20. Thus, it is respectfully submitted that claim 20 is allowable. Because claims 22-33 depend from, and, therefore include all of the limitations of claim 20, it is respectfully submitted that these claims are also allowable.

Claim 1 recites similar limitations to claim 20, as acknowledge by the Examiner. Specifically, claim 1 recites “a feature identifier to determine a feature set of the source file,” “a buffer to assemble the feature set,” and “a feature writer to write the feature set into the target file in the target format.” Therefore, for the reasons discussed above with respect to claim 20, it is respectfully requested that claim 1 also allowable. Because claims 4-19 depend from, and therefore include all the limitations of, claim 1, it is respectfully requested that these claims also allowable.

Claims 34-37 recite similar limitations to claim 20. Specifically, claim 34 recites “providing a buffer to assemble the feature set.” Claim 35 recites “a buffer to assemble the feature set.” Claims 36 and 37 recite “computer readable program code for assembling the

feature set." Therefore, for the reasons discussed above with respect to claim 20, it is respectfully submitted that claims 34-37 are also allowable.

While the Examiner grouped claim 38 with the claims rejected under 35 U.S.C. §102 (e), there is no discussion of claim 38 until the section regarding the 35 U.S.C. § 103 (a) rejection. (*See 12/05/05 Office Action*, p. 7, lines 1-10). Applicant assumes the inclusion of claim 38 with the claims rejected under 35 U.S.C. §102 (e) to be a typographical error. Thus, Applicant only responds to the rejection of claim 38 under 35 U.S.C. §103 (a).

II. The Rejection of Claim 38 Under 35 U.S.C. § 103(a) as Being Obvious Over U.S. Patent Application Pub. No. 2002/0052893 to Grobler et al. Should Be Reversed.

A. The Examiner's Rejection

In the final Office Action, the Examiner rejected claim 38 under 35 U.S.C. § 103 (a) as being unpatentable over the Grobler publication. (*See Id.*).

B. The Grobler Publication Does Not Disclose or Suggest "a Feature Writer Having a Back-End Lookup Table to Map the Feature Set to HTML Code Fragments, to Write the Feature Set into the Target File in the HTML Format" as Recited In Claim 38.

Claim 38 recites a "translator for translating a source file in an MIF format to a target file in an HTML format, the translator comprising: a feature identifier having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file; a buffer to store the feature set; and *a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format.*" The Examiner acknowledged that the Grobler publication fails

to disclose the code fragments of the source file is the MIF code. (*See 12/05/05 Office Action*, p. 7, line 20 – p. 8 line 2). A back-end lookup table is described as a table at the back end, or the output end, that is used by the translator to associate features of a set with various code fragments or tags. (*See Specification*, p. 9, lines 15-24). As described in detail above, a set of features of the present invention is distinct from the tags of the Grobler publication. This distinction is further supported by the fact that the back-end table allows the features to be associated with different tags. If sets of features were equivalent to tags, there would be no need for the table to provide an association between the same components.

The use of the back-end lookup table allows for a feature-reader of the present invention to be used in conjunction with programs that save files in a tree structured language without using a lookup table, such as XML, or with programs that do not use a feature-based reader, such as FrameMaker® HTML. (*See Id.*, p. 10, line 31- p. 11, line 10). Without a lookup table or a feature-base reader, the output is limited to slightly different interpretations of the same output format, or an output having closely related flavors of a generalized format like XML. (*See Id.*). Thus, the back-end lookup table allows the writing function of the present invention to write nominally distinct tree structured formats, sequential formats, post-fix formats and other discrete file format structures. (*See Id.*). The Examiner asserts that the Grobler publication discloses a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format. (*See 12/05/05 Office Action*, p. 7, lines 17-19). However, this assertion is incorrect. The Grobler publication describes the importing of source files into target file and a target table having user-specified characteristics. (*See, Grobler publication, ¶ 0069*). The user can choose to modify the columns and the data, or choose to create a new target table. (*See Id.*). The target table, regardless of

whether or not it is user-specified, does not perform the same functions as the back-end lookup table. Specifically, this target table does not allow for a feature-reader of the present invention to be used in conjunction with programs that save files in a tree structured language.

In view of the above arguments, it is respectfully submitted that the Grobler publication does not teach or suggest “a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format” as recited in claim 38. Accordingly, it is respectfully submitted that claim 38 is allowable over the Grobler publication.

8. Conclusions

For the reasons set forth above, Appellant respectfully requests that the Board reverse the final rejections of the claims by the Examiner under 35 U.S.C. § 102(e) and § 103(a), and indicate that claims 1, 4-20 and 22-38 are allowable.

Respectfully submitted,

By:   
Michael J. Marcin (Reg. No. 48,198)

Date: June 2, 2006

Fay Kaplun & Marcin, LLP  
150 Broadway, Suite 702  
New York, NY 10038  
Tel: (212) 619-6000  
Fax: (212) 619-0276

**CLAIMS APPENDIX**

1. A translator for translating a source file in a source format to a target file in a target format, the translator comprising:
  - a feature identifier to determine a feature set of the source file;
  - a buffer to assemble the feature set; and
  - a feature writer to write the feature set into the target file in the target format.
- 2 (Cancelled)
3. (Cancelled)
4. The translator of claim 1, wherein features of the feature set are selected from the group consisting of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, user-created features, implied feature endings and combinations thereof.
5. The translator of claim 1, wherein the feature identifier comprises a front-end converter to map code fragments of the source file to a list of features.
6. The translator of claim 5, wherein the feature identifier comprises a front-end lookup table.
7. The translator of claim 6, wherein the front-end lookup table is user modifiable.
8. The translator of claim 1, wherein the feature writer comprises a back-end converter to map the feature set to code fragments of the target file format.

9. The translator of claim 8, wherein the back-end converter comprises a back-end lookup table.
10. The translator of claim 5, comprising a plurality of feature writers to write the feature set into a plurality of target files having a plurality of target formats.
11. The translator of claim 1, comprising a plurality of feature identifiers to determine a feature set of a plurality of source files having a plurality of source formats.
12. The translator of claim 5, wherein the front-end converter comprises a lexical analyzer to identify tokens disposed within the source file, and a feature collector to associate the tokens with the feature set.
13. The translator of claim 1, further comprising a user interface.
14. The translator of claim 13, wherein the user interface comprises a GUI.
15. The translator of claim 1, further comprising a source format adapter module to interface with a source file generator.
16. The translator of claim 15, wherein the source format adapter module enables the source file generator to initiate translation by the translator.
17. The translator of claim 1, further comprising a target file adapter module to perform secondary translation.
18. The translator of claim 17, wherein the target file adapter module translates the target file into another target format.
19. The translator of claim 1, wherein the source and target formats are selected from the group consisting of MIF, RTF, WordPerfect, VENTURA, Microsoft Word, Interleaf, HTML,

SGML, SML, C, C++, Visual Basic, Pascal, Java, MFC, PowerPlant, Swing, SVG, HPJ, Flash, WMF, VRML, RenderMan, 3DMF, and combinations thereof.

20. A method of translating a file from a source format to a target format, the method comprising:

- (a) identifying a feature set of a source file;
- (b) assembling the feature set in a buffer; and
- (c) writing the feature set into a target file in the target format.

21. (Cancelled)

22. The method of claim 20, wherein features of the feature set are selected from the group consisting of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, user-created features, implied feature endings and combinations thereof.

23. The method of claim 20, wherein the identifying step (a) comprises mapping code fragments of the source file to a feature list.

24. The method of claim 23, wherein the identifying step (a) comprises looking up the code fragments in a front-end lookup table.

25. The method of claim 24, further comprising permitting the front-end lookup table to be user modifiable.

26. The method of claim 20, wherein the writing step (b) comprises mapping the feature set to code fragments of the target file format.

27. The method of claim 26, wherein the writing step (b) comprises looking up the feature set in a back-end lookup table.
28. The method of claim 20, wherein the writing step (b) comprises writing the feature set into a plurality of target files having a plurality of target formats.
29. The method of claim 20, wherein the identifying step (a) comprises identifying a feature set of a plurality of source files having a plurality of source formats.
30. The method of claim 20, wherein the identifying step (a) comprises identifying tokens disposed within the source file, and associating the tokens with the feature list.
31. The method of claim 20, further comprising using a source file generator to initiate translation by the translator.
32. The method of claim 20, further comprising using a target file adapter module to perform secondary translation.
33. The method of claim 32, wherein the target file adapter module translates the target file into another target format.
34. A method of configuring a system to translate a source file in a source format to a target file in a target format, the method comprising:
  - (a) providing a feature identifier to determine a feature set of the source file;
  - (b) providing a buffer to assemble the feature set; and
  - (c) providing a feature writer to write the feature set into the target file in the target format.
35. (Currently amended) A system for translating a source file in a source format to a target file in a target format, the system comprising:
  - a feature identifier to determine a feature set of the source file;

a buffer to assemble the feature set; and  
a feature writer to write the feature set into the target file in the target format.

36. An article of manufacture for translating a source file in a source format to a target file in a target format, the article of manufacture comprising:

    a computer usable medium having a computer readable program code embodied therein, the computer usable medium having:

        computer readable program code for identifying a feature set of the source file;

        computer readable program code for assembling the feature set; and

        computer readable program code for writing the feature set into the target file in the target format.

37. Computer readable program code for translating a source file in a source format to a target file in a target format, the computer readable program code comprising:

    computer readable program code for identifying a feature set of the source file;

    computer readable program code for assembling the feature set; and

    computer readable program code for writing the feature set into the target file in the target format.

38. A translator for translating a source file in an MIF format to a target file in an HTML format, the translator comprising:

    a feature identifier having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file;

    a buffer to store the feature set; and

    a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format.

Serial No.: 09/766,335  
Group Art Unit: 2178  
Attorney Docket No.: 40101/07301

**EVIDENCE APPENDIX**

No evidence has been entered or relied upon in the present appeal.

**RELATED PROCEEDING APPENDIX**

No decisions have been rendered regarding the present appeal or any proceedings related thereto.



Attorney Docket No. 40101/07301 (2000.034)

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Inventor(s) : Stevens et al.  
Serial No. : 09/854,111 09 / 766, 335  
Filing Date : January 19, 2001  
For : Conversion System for Translating Structured Documents into  
Multiple Target Formats  
  
Group Art Unit: : 2178  
Examiner : Cong-Lac t Huynh

Mail Stop: Appeal Brief - Patent  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

<b>Certificate of Mailing</b>	
I hereby certify that this correspondence is being deposited with U.S. Postal Services as first class mail in an envelope addressed to:	
Mail Stop: Appeal Brief - Patents Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450	
By:	Date: June 2, 2006
Michael J. Marcin, ( Reg. No. 48,198)	

**TRANSMITTAL**

In support to the Notice of Appeal filed April 4, 2006 and the Advisory Action dated April 21, 2006, transmitted herewith please find an Appeal Brief for filing in the above-identified application. Please charge the Credit Card of **Fay Kaplun & Marcin, LLP** in the amount of \$500.00 (PTO-Form 2038 is enclosed herewith). The Commissioner is hereby authorized to charge the **Deposit Account of Fay Kaplun & Marcin, LLP NO. 50-1492** for any additional required fees. A copy of this paper is enclosed for that purpose.

Respectfully submitted,

By:   
Michael J. Marcin, Reg. 48,198

Dated: June 2, 2006

Fay Kaplun & Marcin, LLP  
150 Broadway, Suite 702  
New York, NY 10038  
Tel: (212) 619-6000  
Fax: (212) 619-0276